

Kafka Improvement Proposals

This page describes a proposed Kafka Improvement Proposal (KIP) process for proposing a major change to Kafka.

- [Getting Started](#)
- [Purpose](#)
- [What is considered a "major change" that needs a KIP?](#)
- [What should be included in a KIP?](#)
- [Who should initiate the KIP?](#)
- [Process](#)
- [KIP round-up](#)
- [Adopted KIPs](#)
- [KIPs under discussion](#)
- [Discarded KIPs](#)
- [KIP Discussion Recordings](#)

Getting Started

If this is your first time contributing:

1. Sign up for the Developer mailing list dev@kafka.apache.org . The instructions to sign up are here: <http://kafka.apache.org/contact>
2. Create a wiki ID (<https://cwiki.apache.org/confluence/signup.action>)
3. Create a Jira ID (It's a different system than the wiki) using the ASF Self-serve Portal (<https://selfserve.apache.org/jira-account.html>)
4. Send an email to the dev mailing list (dev@kafka.apache.org) containing your wiki ID and Jira ID requesting permissions to contribute to Apache Kafka.

Purpose

We want to make Kafka a core architectural component for users. We also support a large number of integrations with other tools, systems, and clients. Keeping this kind of usage health requires a high level of compatibility between releases — core architectural elements can't break compatibility or shift functionality from release to release. As a result each new major feature or public api has to be done in a way that we can stick with it going forward.

This means when making this kind of change we need to think through what we are doing as best we can prior to release. And as we go forward we need to stick to our decisions as much as possible. All technical decisions have pros and cons so it is important we capture the thought process that lead to a decision or design to avoid flip-flopping needlessly.

Hopefully we can make these proportional in effort to their magnitude — small changes should just need a couple brief paragraphs, whereas large changes need detailed design discussions.

This process also isn't meant to discourage incompatible changes — proposing an incompatible change is totally legitimate. Sometimes we will have made a mistake and the best path forward is a clean break that cleans things up and gives us a good foundation going forward. Rather this is intended to avoid accidentally introducing half thought-out interfaces and protocols that cause needless heartburn when changed. Likewise the definition of "compatible" is itself squishy: small details like which errors are thrown when are clearly part of the contract but may need to change in some circumstances, likewise performance isn't part of the public contract but dramatic changes may break use cases. So we just need to use good judgement about how big the impact of an incompatibility will be and how big the payoff is.

What is considered a "major change" that needs a KIP?

Any of the following should be considered a major change:

- Any major new feature, subsystem, or piece of functionality
- Any change that impacts the public interfaces of the project

What are the "public interfaces" of the project?

All of the following are public interfaces that people build around:

- Binary log format
- The network protocol and api behavior
- Any class for which the build generates Javadoc. This includes (but is not limited to; look for javadoc `{ }` sections in `build.gradle` for the complete list):
 - `org/apache/kafka/common/serialization`
 - `org/apache/kafka/common`
 - `org/apache/kafka/common/errors`
 - `org/apache/kafka/clients/producer`
 - `org/apache/kafka/clients/consumer` (eventually, once stable)
- Configuration, especially client configuration
- Monitoring
- Command line tools and arguments

Not all compatibility commitments are the same. We need to spend significantly more time on log format and protocol as these break code in lots of clients, cause downtime releases, etc. Public apis are next as they cause people to rebuild code and lead to compatibility issues in large multi-dependency projects (which end up requiring multiple incompatible versions). Configuration, monitoring, and command line tools can be faster and looser — changes here will break monitoring dashboards and require a bit of care during upgrades but aren't a huge burden.

For the most part monitoring, command line tool changes, and configs are added with new features so these can be done with a single KIP.

What should be included in a KIP?

A KIP should contain the following sections:

- **Motivation:** describe the problem to be solved
- **Proposed Change:** describe the new thing you want to do. This may be fairly extensive and have large subsections of its own. Or it may be a few sentences, depending on the scope of the change.
- **New or Changed Public Interfaces:** impact to any of the "compatibility commitments" described above. We want to call these out in particular so everyone thinks about them.
- **Migration Plan and Compatibility:** if this feature requires additional support for a no-downtime upgrade describe how that will work
- **Rejected Alternatives:** What are the other alternatives you considered and why are they worse? The goal of this section is to help people understand why this is the best solution now, and also to prevent churn in the future when old alternatives are reconsidered.

Who should initiate the KIP?

Anyone can initiate a KIP but you shouldn't do it unless you have an intention of getting the work done to implement it (otherwise it is silly).

Process

Here is the process for making a KIP:

1. Click [Create KIP](#). Take the next available KIP number and give your proposal a descriptive heading. e.g. "KIP 42: Allow Infinite Retention With Bounded Disk Usage".
2. Update the next available KIP number below.
3. Fill in the sections as described above
4. Also add an entry to the table [KIPs under discussion](#) (for Streams API KIPs, please also add it to [Kafka Streams sub page](#)).
5. Start a [DISCUSS] thread on the Apache mailing list. Please ensure that the subject of the thread is of the format *[DISCUSS] KIP-{your KIP number} {your KIP heading}* and the body contains a link to your new KIP. The discussion should happen on the mailing list, not on the wiki, since the wiki comment system doesn't work well for larger discussions. In the process of the discussion you may update the proposal. You should let people know the changes you are making. When you feel you have a finalized proposal
6. Once the proposal is finalized call a [VOTE] to have the proposal adopted. These proposals are more serious than code changes and more serious even than release votes. The criteria for acceptance is [lazy majority](#). The vote should remain open for at least 72 hours.
7. **Please** update the KIP wiki page, and the index below, to reflect the current stage of the KIP after a vote. This acts as the permanent record indicating the result of the KIP (e.g., Accepted or Rejected). Also report the result of the KIP vote to the voting thread on the mailing list so the conclusion is clear.e

KIP round-up

Next KIP Number: 1020

Use this number as the identifier for your KIP and increment this value.

Adopted KIPs

Please insert new rows in **sorted order** (descending by KIP number).

KIP (please keep this sorted by KIP number)	Release
KIP-1013: Drop broker and tools support for Java 11 in Kafka 4.0 (deprecate in 3.7)	4.0.0
KIP-1012: The need for a Kafka 3.8.x release	3.8.0
KIP-1007: Introduce Remote Storage Not Ready Exception	3.8.0
KIP-1005: Expose EarliestLocalOffset and TieredOffset	3.8.0
KIP-1004: Enforce tasks.max property in Kafka Connect	3.8.0
KIP-1001: Add CurrentControllerId Metric	3.7.0
KIP-1000: List Client Metrics Configuration Resources	3.7.0
KIP-996: Pre-Vote	TBD (targeting 3.8.0/4.0.0)

KIP-994: Minor Enhancements to ListTransactions and DescribeTransactions APIs	TBD (targeting 3.8.0)
KIP-993: Allow restricting files accessed by File and Directory ConfigProviders	3.8.0
KIP-992: Proposal to introduce IQv2 Query Types: TimestampedKeyQuery and TimestampedRangeQuery	3.7.0
KIP-988: Streams Standby Task Update Listener	3.7.0
KIP-985: Add reverseRange and reverseAll query over kv-store in IQv2	3.7.0
KIP-980: Allow creating connectors in a stopped state	TBD (targeting 3.7.0)
KIP-979: Allow independently stop KRaft processes	TBD (targeting 3.7.0)
KIP-978: Allow dynamic reloading of certificates with different DN / SANs	3.7.0
KIP-977: Partition-level Throughput Metrics	TBD (targeting 3.8.0)
KIP-976: Cluster-wide dynamic log adjustment for Kafka Connect	3.7.0
KIP-975: Docker Image for Apache Kafka	TBD (targeting 3.7.0)
KIP-974: Docker Image for GraalVM based Native Kafka Broker	TBD (targeting 3.7.0)
KIP-970: Deprecate and remove Connect's redundant task configurations endpoint	3.7.0 (deprecation) 4.0.0 (removal)
KIP-968: Support single-key_multi-timestamp Interactive Queries (IQv2) for Versioned State Stores	3.7.0
KIP-966: Eligible Leader Replicas	TBD
KIP-963: Additional metrics in Tiered Storage	3.7.0
KIP-962: Relax non-null key requirement in Kafka Streams	3.7.0
KIP-960: Support single-key_single-timestamp Interactive Queries (IQv2) for Versioned State Stores	3.7.0
KIP-959: Add BooleanConverter to Kafka Connect	3.7.0
KIP-954: expand default DSL store configuration to custom types	3.7.0
KIP-951: Leader discovery optimisations for the client	3.7.0
KIP-949: Add flag to enable the usage of topic separator in MM2 DefaultReplicationPolicy	Adopted, targeting 3.6
KIP-942: Add Power(ppc64le) support	TBD
KIP-941: Range queries to accept null lower and upper bounds	3.6.0
KIP-938: Add more metrics for measuring KRaft performance	TBD
KIP-937: Improve Message Timestamp Validation	3.6.0
KIP-927: Improve the kafka-metadata-quorum output	3.6.0
KIP-925: Rack aware task assignment in Kafka Streams	3.7.0 (completed) 3.6.0 (partially implemented)
KIP-923: Add A Grace Period to Stream Table Join	3.6.0
KIP-919: Allow AdminClient to Talk Directly with the KRaft Controller Quorum and add Controller Registration	3.7.0
KIP-917: Additional custom metadata for remote log segment	3.6.0
KIP-915: Txn and Group Coordinator Downgrade Foundation	3.5.0
KIP-914: DSL Processor Semantics for Versioned Stores	3.5.0
KIP-911: Add source tag to Mirror source metric	3.5.0
KIP-909: DNS Resolution Failure Should Not Fail the Clients	TBD
KIP-907: Add Boolean Serde to public interface	3.5.0

KIP-906: Tools migration guidelines	3.5.0
KIP-904: Kafka Streams - Guarantee subtractor is called before adder if key has not changed	3.5.0
KIP-903: Replicas with stale broker epoch should not be allowed to join the ISR	3.5.0
KIP-902: Upgrade Zookeeper to 3.8.2	3.6.0
KIP-900: KRaft kafka-storage.sh API additions to support SCRAM for Kafka Brokers	3.5.0
KIP-898: Modernize Connect plugin discovery	3.6.0
KIP-896: Remove old client protocol API versions in Kafka 4.0	3.7.0 - new metric & request log attribute 4.0.0 - actual removal
KIP-894: Use incrementalAlterConfig for syncing topic configurations	3.5.0
KIP-893: The Kafka protocol should support nullable structs	3.5.0
KIP-892: Transactional Semantics for StateStores	TBD (targeting 3.7.0)
KIP-890: Transactions Server-Side Defense	TDB
KIP-889: Versioned State Stores	3.5.0
KIP-887: Add ConfigProvider to make use of environment variables	3.5.0
KIP-884: Add config to configure KafkaClientSupplier in Kafka Streams	3.5.0
KIP-881: Rack-aware Partition Assignment for Kafka Consumers	3.4.0 (protocol) 3.5.0 (assignors)
KIP-878: Autoscaling for Stateless & Statically Partitioned Streams	TBD
KIP-876: Time based cluster metadata snapshots	3.4.0
KIP-875: First-class offsets support in Kafka Connect	3.5.0 (STOPPED state, offsets read API) 3.6.0 (offsets alter/reset APIs)
KIP-872: Add Serializer#serializeToByteBuffer() to reduce memory copying	Accepted
KIP-869: Improve Streams State Restoration Visibility	3.5.0 (partially implemented) / TDB
KIP-868 Metadata Transactions	3.6.0
KIP-866 ZooKeeper to KRaft Migration	3.4.0
KIP-865: Support --bootstrap-server in kafka-streams-application-reset	3.4.0
KIP-863: Reduce CompletedFetch#parseRecord() memory copy	TBD
KIP-862: Self-join optimization for stream-stream joins	3.4.0
KIP-859: Add Metadata Log Processing Error Related Metrics	3.3.0
KIP-858: Handle JBOD broker disk failure in KRaft	3.7.0
KIP-854 Separate configuration for producer ID expiry	3.4.0
KIP-852: Optimize calculation of size for log in remote tier	Accepted. Targeting 3.6.0.
KIP-851: Add requireStable flag into ListConsumerGroupOffsetsOptions	3.3.0
KIP-848: The Next Generation of the Consumer Rebalance Protocol	TBD
KIP-847: Add ProducerIdCount metrics	3.5.0
KIP-846: Source/sink node metrics for Consumed/Produced throughput in Streams	3.3.0
KIP-844: Transactional State Stores	TBD
KIP-843: Adding addMetricIfAbsent method to Metrics	3.3.0
KIP-841: Fenced replicas should not be allowed to join the ISR in KRaft	3.3.0

KIP-840: Config file option for MessageReader/MessageFormatter in ConsoleProducer/ConsoleConsumer	3.4.0
KIP-837: Allow MultiCasting a Result Record.	3.4.0
KIP-836: Expose replication information of the cluster metadata	3.3.0
KIP-835: Monitor KRaft Controller Quorum Health	3.3.0
KIP-834: Pause / Resume KafkaStreams Topologies	3.3.0
KIP-833: Mark KRaft as Production Ready	3.3.0
KIP-831: Add metric for log recovery progress	3.3.0
KIP-830: Allow disabling JMX Reporter	3.4.0
KIP-827: Expose logdirs total and usable space via Kafka API	3.3.0
KIP-825: introduce a new API to control when aggregated results are produced	3.3.0
KIP-824: Allowing dumping segmentlogs limiting the batches in the output	3.3.0
KIP-821: Connect Transforms support for nested structures	TBD
KIP-820: Extend KStream process with new Processor API	3.3.0
KIP-815: Support max-timestamp in GetOffsetShell	3.2.0
KIP-814: Static membership protocol should let the leader skip assignment	3.2.0
KIP-813: Shareable State Stores	TBD
KIP-812: Introduce another form of the `KafkaStreams.close()` API that forces the member to leave the consumer group	3.3.0
KIP-811: Add config repartition.purge.interval.ms to Kafka Streams	3.2.0
KIP-810: Allow producing records with null values in Kafka Console Producer	3.2.0
KIP-808: Add support for different unix precisions in TimestampConverter SMT	3.2.0
KIP-806: Add session and window query over kv-store in IQv2	3.2.0
KIP-805: Add range and scan query over kv-store in IQv2	3.2.0
KIP-801: Implement an Authorizer that stores metadata in __cluster__metadata	3.2.0
KIP-800: Add reason to JoinGroupRequest and LeaveGroupRequest	3.2.0
KIP-798: Add possibility to write kafka headers in Kafka Console Producer	3.2.0
KIP-797: Accept duplicate listener on port for IPv4/IPv6	3.6.0
KIP-796: Interactive Query v2	TBD
KIP-794: Strictly Uniform Sticky Partitioner	3.3.0
KIP-793: Allow sink connectors to be used with topic-mutating SMTs	3.6.0
KIP-792: Add "generation" field into consumer protocol	3.4.0
KIP-791: Add Record Metadata to StateStoreContext	3.2.0
KIP-788: Allow configuring num.network.threads per listener	3.2.0
KIP-787: MM2 manage Kafka resources with custom Admin implementation.	3.4.0
KIP-784: Add top-level error code field to DescribeLogDirsResponse	3.2.0
KIP-783: Add TaskId field to StreamsException	3.1.0
KIP-779: Allow Source Tasks to Handle Producer Exceptions	3.2.0
KIP-778: KRaft to KRaft Upgrades	3.3.0
KIP-775: Custom partitioners in foreign key joins	3.1.0
KIP-773: Differentiate consistently metric latency measured in millis and nanos	3.1.0

KIP-771: KRaft brokers without the "controller" role should not expose controller metrics	3.0.0
KIP-770: Replace "buffered.records.per.partition" with "input.buffer.max.bytes"	3.4.0 (WIP)
KIP-769: Connect APIs to list all connector plugins and retrieve their configuration definitions	3.2.0
KIP-768: Extend SASL/OAUTHBEARER with Support for OIDC	3.1.0
KIP-766: fetch/findSessions queries with open endpoints for SessionStore /WindowStore	3.1.0
KIP-764: Configurable backlog size for creating Acceptor	3.2.0
KIP-763: Range Queries with Open Endpoints	3.1.0
KIP-761: Add Total Blocked Time Metric to Streams	3.1.0
KIP-751: Drop support for Scala 2.12 in Kafka 4.0 (deprecate in 3.0)	3.0.0 (deprecation notice), 4.0.0 (support withdrawal)
KIP-750: Drop support for Java 8 in Kafka 4.0 (deprecate in 3.0)	3.0.0 (deprecation notice), 4.0.0 (support withdrawal)
KIP-748: Add Broker Count Metrics	3.1.0
KIP-746: Revise KRaft Metadata Records	3.0.0
KIP-745: Connect API to restart connector and tasks	3.0.0
KIP-744: Migrate TaskMetadata and ThreadMetadata to an interface with internal implementation	3.0.0
KIP-743: Remove config value 0.10.0-2.4 of Streams built-in metrics version config	3.0.0
KIP-741: Change default serde to be null	3.0.0
KIP-740: Clean up public API in TaskId	3.0.0
KIP-738: Removal of Connect's internal converter properties	3.0.0
KIP-735: Increase default consumer session timeout	3.0.0
KIP-734: Improve AdminClient.listOffsets to return timestamp and offset for the record with the largest timestamp	3.0.0
KIP-733: change Kafka Streams default replication factor config	3.0.0
KIP-732: Deprecate eos-alpha and replace eos-beta with eos-v2	3.0.0
KIP-730: Producer ID generation in KRaft mode	3.0.0
KIP-726: Make the "cooperative-sticky, range" as the default assignor	WIP
KIP-725: Streamlining configurations for WindowedSerializer and WindowedDeserializer.	3.0.0
KIP-724: Drop support for message formats v0 and v1	3.0.0 (deprecation) 4.0.0 (removal)
KIP-722: Enable connector client overrides by default	3.0.0
KIP-721: Enable connector log contexts in Connect Log4j configuration	3.0.0
KIP-720: Deprecate MirrorMaker v1	3.0.0
KIP-719: Deprecate Log4J Appender	WIP
KIP-716: Allow configuring the location of the offset-syncs topic with MirrorMaker2	3.0.0
KIP-715: Expose Committed offset in streams	3.0.0
KIP-714: Client metrics and observability	3.7.0
KIP-710: Full support for distributed mode in dedicated MirrorMaker 2.0 clusters	3.5.0
KIP-709: Extend OffsetFetch requests to accept multiple group ids.	3.3.0
KIP-708: Rack aware StandbyTask assignment for Kafka Streams	3.2.0

KIP-707: The future of KafkaFuture	3.0.0
KIP-704: Send a hint to the partition leader to recover the partition	3.2.0
KIP-700: Add Describe Cluster API	2.8.0
KIP-699: Update FindCoordinator to resolve multiple Coordinators at a time	3.0.0
KIP-698: Add Explicit User Initialization of Broker-side State to Kafka Streams	WIP / inactive
KIP-696: Update Streams FSM to clarify ERROR state meaning	2.8.0
KIP-695: Further Improve Kafka Streams Timestamp Synchronization	3.0.0
KIP-691: Enhance Transactional Producer Exception Handling	WIP / inactive
KIP-690: Add additional configuration to control MirrorMaker 2 internal topics naming convention	3.1.0
KIP-679: Producer will enable the strongest delivery guarantee by default	3.0.0
KIP-689: Extend `StreamJoined` to allow more store configs	2.8.0
KIP-684 - Support mutual TLS authentication on SASL_SSL listeners	2.8.0
KIP-680: TopologyTestDriver should not require a Properties argument	2.8.0
KIP-676: Respect logging hierarchy	2.8.0
KIP-673: Emit JSONs with new auto-generated schema	2.8.0
KIP-671: Introduce Kafka Streams Specific Uncaught Exception Handler	2.8.0
KIP-666: Add Instant-based methods to ReadOnlySessionStore	3.0.0
KIP-663: API to Start and Shut Down Stream Threads	2.8.0
KIP-662: Throw Exception when Source Topics of a Streams App are Deleted	2.7.0
KIP-661: Expose task configurations in Connect REST API	2.8.0
KIP-659: Improve TimeWindowedDeserializer and TimeWindowedSerde to handle window size	2.8.0
KIP-654: Aborted transaction with non-flushed data should throw a non-fatal exception	2.7.0
KIP-653: Upgrade log4j to log4j2	3.2.0 (WIP)
KIP-651 - Support PEM format for SSL certificates and private key	2.7.0
KIP-648 Renaming getter method for Interactive Queries	2.7.0
KIP-641 An new java interface to replace 'kafka.common.MessageReader'	3.5.0
KIP-635: GetOffsetShell: support for multiple topics and consumer configuration override	3.0
KIP-633: Deprecate 24-hour Default Grace Period for Windowed Operations in Streams	3.0.0
KIP-632: Add DirectoryConfigProvider	2.7.0
KIP-631: The Quorum-based Kafka Controller	2.8.0
KIP-630: Kafka Raft Snapshot	3.0.0
KIP-629: Use racially neutral terms in our codebase	3.0.0
KIP-627: Expose Trogdor-specific JMX Metrics for Tasks and Agents	WIP
KIP-626: Rename StreamsConfig config variable name	2.7.0
KIP-623: Add "internal-topics" option to streams application reset tool	3.0.0
KIP-622: Add currentTimeMs and currentStreamTimeMs to ProcessorContext	3.0.0
KIP-621: Deprecate and replace DescribeLogDirsResult.all() and .values()	2.7.0

KIP-620 Deprecate ConsumerConfig#addDeserializerToConfig and ProducerConfig#addSerializerToConfig	2.7.0
KIP-618: Exactly-Once Support for Source Connectors	3.3.0 (almost everything) 3.2.0 (new admin client API)
KIP-617: Allow Kafka Streams State Stores to be iterated backwards	2.7.0
KIP-616: Rename implicit Serdes instances in kafka-streams-scala	2.7.0
KIP-614: Add Prefix Scan support for State Stores	2.8.0
KIP-613: Add end-to-end latency metrics to Streams	2.6.0 (INFO node-level metrics), remainder 2.7.0 (TRACE store-level metrics)
KIP-612: Ability to Limit Connection Creation Rate on Brokers	2.7.0 / 2.8.0
KIP-610: Error Reporting in Sink Connectors	2.6.0
KIP-608 - Expose Kafka Metrics in Authorizer	WIP
KIP-607: Add Metrics to Kafka Streams to Report Properties of RocksDB	2.7.0
KIP-606: Add Metadata Context to MetricsReporter	2.6.0
KIP-605: Expand Connect Worker Internal Topic Settings	2.6.0
KIP-604: Remove ZooKeeper Flags from the Administrative Tools	one part in 2.6.0, most parts scheduled for 3.0
KIP-602: Change default value for client.dns.lookup	2.6.0
KIP-601: Configurable socket connection timeout	2.7.0
KIP-599: Throttle Create Topic, Create Partition and Delete Topic Operations	2.7.0
KIP-597: MirrorMaker2 internal topics Formatters	2.7.0
KIP-594: Expose output topic names from TopologyTestDriver	2.6.0
KIP-591: Add Kafka Streams config to set default state store	3.2.0 (WIP)
KIP-590: Redirect Zookeeper Mutation Protocols to The Controller	2.8.0 (WIP)
KIP-589 Add API to update Replica state in Controller	2.8.0 (WIP)
KIP-588: Allow producers to recover gracefully from transaction timeouts	2.8.0 (WIP)
KIP-586: Deprecate commit records without record metadata	2.6.0
KIP-585: Filter and Conditional SMTs	2.6.0
KIP-584: Versioning scheme for features	2.7.0
KIP-581: Value of optional null field which has default value	3.5.0
KIP-580: Exponential Backoff for Kafka Clients	3.7.0
KIP-577: Allow HTTP Response Headers to be Configured for Kafka Connect	2.6.0
KIP-574: CLI Dynamic Configuration with file input	2.6.0
KIP-573: Enable TLSv1.3 by default	2.6.0
KIP-572: Improve timeouts and retries in Kafka Streams	2.8.0 (partially implemented in 2.7.0)
KIP-571: Add option to force remove members in StreamsResetter	2.6.0
KIP-570: Add leader epoch in StopReplicaRequest	2.6.0
KIP-569: DescribeConfigsResponse - Update the schema to include additional metadata information of the field	2.6.0
KIP-568: Explicit rebalance triggering on the Consumer	2.6.0
KIP-562: Allow fetching a key from a single partition rather than iterating over all the stores on an instance	2.5.0
KIP-559: Make the Kafka Protocol Friendlier with L7 Proxies	2.5.0

KIP-558: Track the set of actively used topics by connectors in Kafka Connect	2.5.0
KIP-557: Add emit on change support for Kafka Streams	Implemented for KTable source nodes in 2.6.0, full implementation planned for subsequent releases.
KIP-555: Deprecate direct Zookeeper access in Kafka administrative tools	2.5.0
KIP-554: Add Broker-side SCRAM Config API	2.7.0
KIP-553: Disable all SSL protocols except TLSV1.2 by default.	2.5.0
KIP-551: Expose disk read and write metrics	2.6.0
KIP-546: Add Client Quota APIs to the Admin Client	2.6.0 (describe/alter), remainder pending (resolve)
KIP-545: support automated consumer offset sync across clusters in MM 2.0	2.7.0
KIP-544: Make metrics exposed via JMX configurable	2.5.0
KIP-543: Expand ConfigCommand's non-ZK functionality	2.5.0
KIP-541: Create a fetch.max.bytes configuration for the broker	2.5.0
KIP-538: Add a metric tracking the number of open connections with a given SSL cipher type	2.5.0
KIP-537: Increase default zookeeper session timeout	2.5.0
KIP-535: Allow state stores to serve stale reads during rebalance	2.5.0
KIP-534: Reorganize checkpoint system in log cleaner to per partition	3.1.0
KIP-533: Add default api timeout to AdminClient	2.5.0
KIP-532: Broker Consumer Lag metrics in size and time	WIP / inactive
KIP-531: Drop support for Scala 2.11 in Kafka 2.5	Initial version in 2.5.0, remainder in later release (WIP)
KIP-530: Consider renaming 'UsePreviousTimeOnInvalidTimeStamp' class to 'UsePartitionTimeOnInvalidTimeStamp'	2.5.0
KIP-528: Deprecate PartitionGrouper configuration and interface	2.4.0
KIP-527: Add VoidSerde to Serdes	2.5.0
KIP-526: Reduce Producer Metadata Lookups for Large Number of Topics	2.5.0
KIP-525 - Return topic metadata and configs in CreateTopics response	2.4.0
KIP-524: Allow users to choose config source when describing configs	2.5.0
KIP-523: Add KStream#toTable to the Streams DSL	2.5.0
KIP-521: Enable redirection of Connect's log4j messages to a file by default	2.4.0
KIP-519: Make SSL context/engine configuration extensible	2.6.0
KIP-518: Allow listing consumer groups per state	2.6.0
KIP-517: Add consumer metrics to observe user poll behavior	2.4.0
KIP-516: Topic Identifiers & Topic Deletion State Improvements	Some in 2.8.0, remainder WIP
KIP-515: Enable ZK client to use the new TLS supported authentication	2.5.0
KIP-514: Add a bounded flush() API to Kafka Producer	2.5.0
KIP-511: Collect and Expose Client's Name and Version in the Brokers	2.4.0 (protocol) / 2.5.0 (metric)
KIP-507: Securing Internal Connect REST Endpoints	2.4.0
KIP-504 - Add new Java Authorizer Interface	2.4.0
KIP-503: Add metric for number of topics marked for deletion	2.4.0
KIP-501: Avoid out-of-sync or offline partitions when follower fetch requests not processed in time	WIP
KIP-500: Replace ZooKeeper with a Self-Managed Metadata Quorum	

KIP-499 - Unify connection name flag for command line tool	2.5.0
KIP-497: Add inter-broker API to alter ISR	2.7.0
KIP-496: Administrative API to delete consumer offsets	2.4.0
KIP-495: Dynamically Adjust Log Levels in Connect	2.4.0
KIP-492: Add java security providers in Kafka Security config	2.4.0
KIP-488: Clean up Sum,Count,Total Metrics	2.4.0
KIP-484: Expose metrics for group and transaction metadata loading duration	2.4.0
KIP-482: The Kafka Protocol should Support Optional Tagged Fields	2.4.0
KIP-481: SerDe Improvements for Connect Decimal type in JSON	2.4.0
KIP-480: Sticky Partitioner	2.4.0
KIP-479: Add StreamJoined config object to Join	2.4.0
KIP-478 - Strongly typed Processor API	2.7.0
KIP-476: Add Java AdminClient Interface	2.4.0
KIP-475: New Metrics to Measure Number of Tasks on a Connector	2.4.0
KIP-474: To deprecate WindowStore#put(key, value)	2.4.0
KIP-471: Expose RocksDB Metrics in Kafka Streams	2.4.0 (metrics blocked due to RocksDB version added in 3.2.0)
KIP-470: TopologyTestDriver test input and output usability improvements	2.4.0
KIP-467: Augment ProduceResponse error messaging for specific culprit records	2.4.0 (partially implemented) / 2.5.0
KIP-466: Add support for List<T> serialization and deserialization	3.0.0
KIP-465: Add Consolidated Connector Endpoint to Connect REST API	2.3.0
KIP-464: Defaults for AdminClient#createTopic	2.4.0
KIP-462: Use local thread id for KStreams	2.3.0
KIP-461: Improve Replica Fetcher behavior at handling partition failure	2.3.0
KIP-460: Admin Leader Election RPC	2.4.0
KIP-458: Connector Client Config Override Policy	2.3.0
KIP-455: Create an Administrative API for Replica Reassignment	2.4.0 (API/broker logic) 2.5.0 (tools changes)
KIP-454: Expansion of the ConnectClusterState interface	2.3.0
KIP-453: Add close() method to RocksDBConfigSetter	2.3.0
KIP-450: Sliding Window Aggregations in the DSL	2.7.0
KIP-449: Add connector contexts to log messages in Connect workers	2.3.0
KIP-447: Producer scalability for exactly once semantics	2.5.0 (broker, consumer, producer changes implemented) / 2.6.0
KIP-446: Add changelog topic configuration to KTable suppress	2.6.0
KIP-445: In-memory Session Store	2.3.0
KIP-444: Augment metrics for Kafka Streams	2.4.0 (partially implemented) / 2.5.0 (partially implemented) / 2.6.0
KIP-443: Return to default segment.ms and segment.index.bytes in Streams repartition topics	2.3.0
KIP-442: Return to default max poll interval in Streams	2.3.0
KIP-441: Smooth Scaling Out for Kafka Streams	2.6.0
KIP-440: Extend Connect Converter to support headers	2.4.0
KIP-437: Custom replacement for MaskField SMT	2.6.0
KIP-436: Add a metric indicating start time	2.3.0

KIP-434: Add Replica Fetcher and Log Cleaner Count Metrics	2.4.0
KIP-431: Support of printing additional ConsumerRecord fields in DefaultMessageFormatter	2.7.0
KIP-430 - Return Authorized Operations in Describe Responses	2.3.0
KIP-429: Kafka Consumer Incremental Rebalance Protocol	2.4.0
KIP-428: Add in-memory window store	2.3.0
KIP-427: Add AtMinIsr topic partition category (new metric & TopicCommand option)	2.3.0
KIP-425: Add some Log4J Kafka Appender Properties for Producing to Secured Brokers	2.3.0
KIP-421: Support resolving externalized secrets in AbstractConfig	2.3.0
KIP-420: Add Single Value Fetch in Session Stores	2.2.0
KIP-418: A method-chaining way to branch KStream	2.8.0
KIP-417: Allow JmxTool to connect to a secured RMI port	2.3.0
KIP-415: Incremental Cooperative Rebalancing in Kafka Connect	2.3.0
KIP-414: Expose Embedded ClientIds in Kafka Streams	2.2.0
KIP-412: Extend Admin API to support dynamic application log levels	2.4.0
KIP-411: Make default Kafka Connect worker task client IDs distinct	2.3.0
KIP-405: Kafka Tiered Storage	Accepted
KIP-402: Improve fairness in SocketServer processors	2.2.0 (partially implemented) / 2.3.0
KIP-401: TransformerSupplier/ProcessorSupplier StateStore connecting	2.6.0
KIP-399: Extend ProductionExceptionHandler to cover serialization exceptions	3.5.0
KIP-396: Add Commit/List Offsets Operations to AdminClient	2.5.0
KIP-394: Require member.id for initial join group request	2.2.0
KIP-393: Time windowed serde to properly deserialize changelog input topic	2.2.0
KIP-392: Allow consumers to fetch from closest replica	2.4.0
KIP-390: Support Compression Level	WIP
KIP-389: Introduce a configurable consumer group size limit	2.2.0
KIP-386: Standardize on Min/Avg/Max metrics' default value	2.2.0
KIP-382: MirrorMaker 2.0	2.4.0
KIP-379: Multiple Consumer Group Management	2.4.0
KIP-380: Detect outdated control requests and bounced brokers using broker generation	2.2.0
KIP-377: TopicCommand to use AdminClient	2.2.0
KIP-376: Implement AutoClosable on appropriate classes that want to be used in a try-with-resource statement	2.2.0
KIP-374: Add '--help' option to all available Kafka CLI commands	2.2.0
KIP-373: Allow users to create delegation tokens for other users	3.3.0
KIP-372: Naming Repartition Topics for Joins and Grouping	2.1.0
KIP-371: Add a configuration to build custom SSL principal name	2.2.0
KIP-369: Alternative Partitioner to Support "Always Round-Robin" Selection	2.4.0
KIP 368: Allow SASL Connections to Periodically Re-Authenticate	2.2.0

KIP-367 Introduce close(Duration) to Producer and AdminClient instead of close(long, TimeUnit)	2.2.0
KIP-366: Make FunctionConversions deprecated	2.1.0
KIP-365: Materialized, Serialized, Joined, Consumed and Produced with implicit Serde	2.1.0
KIP-361: Add Consumer Configuration to Disable Auto Topic Creation	2.3.0
KIP-360: Improve reliability of idempotent/transactional producer	2.4.0 (partially implemented) / 2.5.0
KIP-359: Verify leader epoch in produce requests	2.8.0 (WIP)
KIP-358: Migrate Streams API to Duration instead of long ms times	2.1.0
KIP-357: Add support to list ACLs per principal	2.1.0
KIP-356: Add withCachingDisabled() to StoreBuilder	2.1.0
KIP-354: Add a Maximum Log Compaction Lag	2.3.0
KIP-353: Improve Kafka Streams Timestamp Synchronization	2.1.0
KIP-352: Distinguish URPs caused by reassignment	2.5.0
KIP-351: Add --under-min-isr option to describe topics command	2.3.0
KIP-346: Improve LogCleaner behavior on error	2.1.0
KIP-345: Introduce static membership protocol to reduce consumer rebalances	2.4.0
KIP-342 Add support for custom SASL extensions in OAuthBearer authentication	2.1.0
KIP-341: Update Sticky Assignor's User Data Protocol	2.3.0
KIP-340: Allow kafka-reassign-partitions.sh and kafka-log-dirs.sh to take admin client property file	2.1.0
KIP-339: Create a new IncrementalAlterConfigs API	2.3.0
KIP-338 Support to exclude the internal topics in kafka-topics.sh command	2.1.0
KIP-336: Consolidate ExtendedSerializer/Serializer and ExtendedDeserializer/Deserializer	2.1.0
KIP-332: Update AclCommand to use AdminClient API	2.1.0
KIP-331 Add default implementation to close() and configure() for Serializer, Deserializer and Serde	2.3.0
KIP-330: Add retentionPeriod in SessionBytesStoreSupplier	2.1.0
KIP-328: Ability to suppress updates for KTables	2.1.0 (partially implemented; inactive)
KIP-324: Add method to get metrics() in AdminClient	2.1.0
KIP-322: Return new error code for DeleteTopics API when topic deletion disabled.	2.1.0
KIP-321: Update TopologyDescription to better represent Source and Sink Nodes	2.1.0
KIP-320: Allow fetchers to detect and handle log truncation	2.1.0 (partially implemented). full implementation in 2.4.0
KIP-319: Replace segments with segmentInterval in WindowBytesStoreSupplier	2.1.0
KIP-313: Add KStream.flatTransform and KStream.flatTransformValues	2.2.0 (partially implemented) / 2.3.0
KIP-312 Add Overloaded StreamsBuilder Build Method to Accept java.util.Properties	2.1.0
KIP-309: Add toUpperCase support to sasl.kerberos.principal.to.local rule	2.4.0
KIP-308: Support dynamic update of max.connections.per.ip/max.connections.per.ip.overrides	2.1.0
KIP-307: Allow to define custom processor names with KStreams DSL	2.3.0 (partially implemented) / 2.4.0
KIP-306: Configuration for Delaying Response to Failed Authentication	2.1.0

KIP-305: Add Connect primitive number converters	2.0.0
KIP-303: Add Dynamic Routing in Streams Sink	2.0.0
KIP-302 - Enable Kafka clients to use all DNS resolved IP addresses	2.1.0
KIP-300: Add Windowed KTable API in StreamsBuilder	not implemented; inactive
KIP-298: Error Handling in Connect	2.0.0
KIP-297: Externalizing Secrets for Connect Configurations	2.0.0
KIP-295 Add Streams Configuration Allowing for Optional Topology Optimization	2.0.0
KIP-294 - Enable TLS hostname verification by default	2.0.0
KIP-291: Separating controller connections and requests from the data plane	2.2.0
KIP-292: Add transformValues() method to KTable	2.0.0
KIP-290: Support for Prefixed ACLs	2.0.0
KIP-289: Improve the default group id behavior in KafkaConsumer	2.2.0
KIP-285: Connect Rest Extension Plugin	2.0.0
KIP-284: Set default retention ms for Streams repartition topics to Long.MAX_VALUE	2.0.0
KIP-283: Efficient Memory Usage for Down-Conversion	2.0.0
KIP-282: Add the listener name to the authentication context	2.0.0
KIP-281: ConsumerPerformance: Increase Polling Loop Timeout and Make It Reachable by the End User	2.0.0
KIP-280: Enhanced log compaction	WIP
KIP-279: Fix log divergence between leader and follower after fast leader fail over	2.0.0
KIP-278 - Add version option to Kafka's commands	2.0.0
KIP-277 - Fine Grained ACL for CreateTopics API	2.0.0
KIP-276 - Add StreamsConfig prefix for different consumers	2.0.0
KIP-274: Kafka Streams Skipped Records Metrics	2.0.0
KIP-272: Add API version tag to broker's RequestsPerSec metric	2.0.0
KIP-270 - A Scala Wrapper Library for Kafka Streams	2.0.0
KIP-268: Simplify Kafka Streams Rebalance Metadata Upgrade	2.0.0
KIP-267: Add Processor Unit Test Support to Kafka Streams Test Utils	2.0.0
KIP-266: Fix consumer indefinite blocking behavior	2.0.0
KIP-265: Make Windowed Serde to public APIs	2.0.0
KIP-261: Add Single Value Fetch in Window Stores	2.0.0
KIP-258: Allow to Store Record Timestamps in RocksDB	2.3.0 (partially implemented; inactive)
KIP-257 - Configurable Quota Management	2.0.0
KIP-255: OAuth Authentication via SASL/OAUTHBEARER	2.0.0
KIP-251: Allow timestamp manipulation in Processor API	2.0.0
KIP-249: Add Delegation Token Operations to KafkaAdminClient	2.0.0
KIP-247: Add public test utils for Kafka Streams	1.1.0
KIP-245: Use Properties instead of StreamsConfig in KafkaStreams constructor	2.0.0
KIP-244: Add Record Header support to Kafka Streams Processor API	2.0.0
KIP-243: Make ProducerConfig and ConsumerConfig constructors public	1.1.0
KIP-239 Add queryableStoreName() to GlobalKTable	1.1.0

KIP-238: Expose Kafka cluster ID in Connect REST API	1.1.0
KIP-237: More Controller Health Metrics	2.0.0
KIP-235: Add DNS alias support for secured connection	2.1.0
KIP-233: Simplify StreamsBuilder#addGlobalStore	1.1.0
KIP-231: Improve the Required ACL of ListGroups API	2.1.0
KIP-229: DeleteGroups API	1.1.0
KIP-227 - Introduce Incremental FetchRequests to Increase Partition Scalability	1.1.0
KIP-226 - Dynamic Broker Configuration	1.1.0
KIP-225 - Use tags for consumer "records.lag" metrics	1.1.0
KIP-224: Add configuration parameter `retries` to Streams API	1.1.0
KIP-223 - Add per-topic min lead and per-partition lead metrics to KafkaConsumer	2.0.0
KIP-222 - Add Consumer Group operations to Admin API	2.0.0
KIP-221: Enhance DSL with Connecting Topic Creation and Repartition Hint	2.6.0
KIP-220: Add AdminClient into Kafka Streams' ClientSupplier	1.1.0
KIP-219: Improve quota communication	2.0.0
KIP-218: Make KafkaFuture.Function java 8 lambda compatible	1.1.0
KIP-216: IQ should throw different exceptions for different errors	2.8.0 (WIP)
KIP-215: Add topic regex support for Connect sinks	1.1.0
KIP-214: Add zookeeper.max.in.flight.requests config to the broker	1.1.0
KIP-213 Support non-key joining in KTable	2.4.0
KIP-212: Enforce set of legal characters for connector names	1.1.0
KIP-211: Revise Expiration Semantics of Consumer Group Offsets	2.1.0
KIP-210 - Provide for custom error handling when Kafka Streams fails to produce	1.1.0
KIP-208: Add SSL support to Kafka Connect REST interface	1.1.0
KIP-207: Offsets returned by ListOffsetsResponse should be monotonically increasing even during a partition leader change	2.2.0
KIP-206: Add support for UUID serialization and deserialization	2.1.0
KIP-205: Add all() and range() API to ReadOnlyWindowStore	1.1.0
KIP-204 : Adding records deletion operation to the new Admin Client API	1.1.0
KIP-203: Add toLowerCase support to sasl.kerberos.principal.to.local rule	1.1.0
KIP-202: Move merge() from StreamsBuilder to KStream	1.0.0
KIP-198: Remove ZK dependency from Streams Reset Tool	1.0.0
KIP-197 Connect REST API should include the connector type when describing a connector	1.0.0
KIP-196: Add metrics to Kafka Connect framework	1.0.0
KIP-195: AdminClient.createPartitions	1.0.0
KIP-192 : Provide cleaner semantics when idempotence is enabled	1.0.0
KIP-191: KafkaConsumer.subscribe() overload that takes just Pattern	1.0.0
KIP-190: Handle client-ids consistently between clients and brokers	1.0.0
KIP-189: Improve principal builder interface and add support for SASL	1.0.0
KIP-188 - Add new metrics to support health checks	1.0.0
KIP-187 - Add cumulative count metric for all Kafka rate metrics	1.0.0

KIP-186: Increase offsets retention default to 7 days	2.0.0
KIP-183 - Change PreferredReplicaLeaderElectionCommand to use AdminClient	2.2.0
KIP-182: Reduce Streams DSL overloads and allow easier use of custom storage engines	1.0.0
KIP-180: Add a broker metric specifying the number of consumer group rebalances in progress	1.1.0
KIP-177: Consumer perf tool should count rebalance time	1.0.0
KIP-176: Remove deprecated new-consumer option for tools	2.0.0
KIP-175: Additional '--describe' views for ConsumerGroupCommand	1.1.0
KIP-174 - Deprecate and remove internal converter configs in WorkerConfig	2.0.0
KIP-173: Add prefix to StreamsConfig to enable setting default internal topic configs	1.0.0
KIP-171 - Extend Consumer Group Reset Offset for Stream Application	1.1.0
KIP-168: Add GlobalTopicCount and GlobalPartitionCount metric per cluster	1.0.0
KIP-167: Add interface for the state store restoration process	1.0.0
KIP-164 - Add UnderMinIsrPartitionCount and per-partition UnderMinIsr metrics	1.0.0
KIP-163: Lower the Minimum Required ACL Permission of OffsetFetch	1.0.0
KIP-162: Enable topic deletion by default	1.0.0
KIP-161: streams deserialization exception handlers	1.0.0
KIP-160: Augment KStream.print(), KStream.writeAsText() to allow users pass in extra parameters in the printed string	1.0.0
KIP-158: Kafka Connect should allow source connectors to set topic-specific settings for new topics	2.6.0
KIP-157 - Add consumer config options to streams reset tool	1.0.0
KIP-156 Add option "dry run" to Streams application reset tool	0.11.0.0
KIP-155 - Add range scan for windowed state stores	0.11.0.0
KIP-154 Add Kafka Connect configuration properties for creating internal topics	0.11.0.0
KIP-153: Include only client traffic in BytesOutPerSec metric	0.11.0.0
KIP-152 - Improve diagnostics for SASL authentication failures	1.0.0
KIP-151 Expose Connector type in REST API	0.11.0.0
KIP-150 - Kafka-Streams Cogroup	2.5.0
KIP-149: Enabling key access in ValueTransformer, ValueMapper, and ValueJoiner	1.1.0 (partially implemented; inactive)
KIP-146 - Classloading Isolation in Connect	0.11.0.0
KIP-145 - Expose Record Headers in Kafka Connect	1.1.0
KIP-144: Exponential backoff for broker reconnect attempts	0.11.0.0
KIP-143: Controller Health Metrics	0.11.0.0
KIP-140: Add administrative RPCs for adding, deleting, and listing ACLs	0.11.0.0
KIP-138: Change punctuate semantics	1.0.0
KIP-137: Enhance TopicCommand --describe to show topics marked for deletion	0.11.0.0
KIP-136: Add Listener name to SelectorMetrics tags	0.11.0.0
KIP-134: Delay initial consumer group rebalance	0.11.0.0
KIP-133: Describe and Alter Configs Admin APIs	0.11.0.0

KIP-131 - Add access to OffsetStorageReader from SourceConnector	2.6.0
KIP-130: Expose states of active tasks to KafkaStreams public API	1.0.0
KIP-129: Streams Exactly-Once Semantics	0.11.0.0
KIP-128: Add ByteArrayConverter for Kafka Connect	0.11.0.0
KIP-126 - Allow KafkaProducer to split and resend oversized batches.	0.11.0.0
KIP-124 - Request rate quotas	0.11.0.0
KIP-123: Allow per stream/table timestamp extractor	0.11.0.0
KIP-122: Add Reset Consumer Group Offsets tooling	0.11.0.0
KIP-121: Add KStream peek method	0.11.0.0
KIP-120: Cleanup Kafka Streams builder API	1.0.0
KIP-119: Drop Support for Scala 2.10 in Kafka 0.11	0.11.0.0
KIP-118: Drop Support for Java 7	2.0.0
KIP-117: Add a public AdminClient API for Kafka admin operations	0.11.0.0
KIP-115: Enforce offsets.topic.replication.factor upon __consumer__offsets auto topic creation	0.11.0.0
KIP-114: KTable state stores and improved semantics	0.11.0.0
KIP-113: Support replicas movement between log directories	1.1.0
KIP-112: Handle disk failure for JBOD	1.0.0
KIP-110: Add Codec for ZStandard Compression	2.1.0
KIP-109: Old Consumer Deprecation	0.11.0.0
KIP-108: Create Topic Policy	0.10.2.0
KIP-107: Add deleteRecordsBefore() API in AdminClient	0.11.0.0
KIP-106 - Change Default unclean.leader.election.enabled from True to False	0.11.0.0
KIP-105: Addition of Record Level for Sensors	0.10.2.0
KIP-104: Granular Sensors for Streams	0.10.2.0
KIP-103: Separation of Internal and External traffic	0.10.2.0
KIP-102 - Add close with timeout for consumers	0.10.2.0
KIP-101 - Alter Replication Protocol to use Leader Epoch rather than High Watermark for Truncation	0.11.0.0
KIP-100 - Relax Type constraints in Kafka Streams API	0.10.2.0
KIP-99: Add Global Tables to Kafka Streams	0.10.2.0
KIP-98 - Exactly Once Delivery and Transactional Messaging	0.11.0.0
KIP-97: Improved Kafka Client RPC Compatibility Policy	0.10.2.0
KIP-96 - Add per partition metrics for in-sync and assigned replica count	0.10.2.0
KIP-94 Session Windows	0.10.2.0
KIP-93: Improve invalid timestamp handling in Kafka Streams	0.10.2.0
KIP-92 - Add per partition lag metrics to KafkaConsumer	0.10.2.0
KIP-91 Provide Intuitive User Timeouts in The Producer	2.1.0
KIP-90 - Remove zkClient dependency from Streams	0.10.2.0
KIP-89: Allow sink connectors to decouple flush and offset commit	0.10.2.0
KIP-88: OffsetFetch Protocol Update	0.10.2.0
KIP-86: Configurable SASL callback handlers	2.0.0

KIP-85: Dynamic JAAS configuration for Kafka clients	0.10.2.0
KIP-84: Support SASL SCRAM mechanisms	0.10.2.0
KIP-82 - Add Record Headers	0.11.0.0
KIP-81: Bound Fetch memory usage in the consumer	2.8.0 (WIP)
KIP-79 - ListOffsetRequest/ListOffsetResponse v1 and add timestamp search methods to the new consumer	0.10.1.0
KIP-78: Cluster Id	0.10.1.0
KIP-77: Improve Kafka Streams Join Semantics	0.10.2.0
KIP-75 - Add per-connector Converters	0.10.1.0
KIP-74: Add Fetch Response Size Limit in Bytes	0.10.1.0
KIP-73: Replication Quotas	0.10.1.0
KIP-72: Allow putting a bound on memory consumed by Incoming request	1.0.0
KIP-71: Enable log compaction and deletion to co-exist	0.10.1.0
KIP-70: Revise Partition Assignment Semantics on New Consumer's Subscription Change	0.10.1.0
KIP-67: Queryable state for Kafka Streams	0.10.1.0
KIP-66: Single Message Transforms for Kafka Connect	0.10.2.0 / 0.11.0.0
KIP-65: Expose timestamps to Connect	0.10.1.0
KIP-63: Unify store and downstream caching in streams	0.10.1.0
KIP-62: Allow consumer to send heartbeats from a background thread	0.10.1.0
KIP-60 - Make Java client classloading more flexible	0.10.1.0
KIP-58 - Make Log Compaction Point Configurable	0.10.1.0
KIP-57 - Interoperable LZ4 Framing	0.10.0.0
KIP-56: Allow cross origin HTTP requests on all HTTP methods	0.10.0.0
KIP-55: Secure Quotas for Authenticated Users	0.10.1.0
KIP-54: Sticky Partition Assignment Strategy	0.11.0.0
KIP-52: Connector Control APIs	0.10.0.0
KIP-51 - List Connectors REST API	0.10.0.0
KIP-48 Delegation token support for Kafka	1.1.0
KIP-45 - Standardize all client sequence interaction on j.u.Collection.	0.10.0.0
KIP-43: Kafka SASL enhancements	0.10.0.0
KIP-42: Add Producer and Consumer Interceptors	0.10.0.0
KIP-41: Consumer Max Records	0.10.0.0
KIP-40: ListGroups and DescribeGroup	0.9.0.0
KIP-38: ZooKeeper Authentication	0.9.0.0
KIP-36 - Rack aware replica assignment	0.10.0.0
KIP-35 - Retrieving protocol version	0.10.0.0
KIP-33 - Add a time based log index	0.10.1.0
KIP-32 - Add timestamps to Kafka message	0.10.0.0
KIP-31 - Move to relative offsets in compressed message sets	0.10.0.0
KIP-28 - Add a processor client	0.10.0.0
KIP-26 - Add Kafka Connect framework for data import/export	0.9.0.0

KIP-25 - System test improvements	0.9.0.0
KIP-22 - Expose a Partitioner interface in the new producer	0.9.0.0
KIP-21 - Dynamic Configuration	0.9.0.0 (WIP)
KIP-20 Enable log preallocate to improve consume performance under windows and some old Linux file system	0.9.0.0
KIP-19 - Add a request timeout to NetworkClient	0.9.0.0
KIP-16 - Automated Replica Lag Tuning	0.9.0.0
KIP-15 - Add a close method with a timeout in the producer	0.9.0.0
KIP-13 - Quota Design	0.9.0.0
KIP-12 - Kafka Sasl/Kerberos and SSL implementation	0.9.0.0
KIP-11 - Kafka Authorizer design	0.9.0.0
KIP-8 - Add a flush method to the producer API	0.9.0.0
KIP-4 - Metadata Protocol Changes	0.10.0.0
KIP-4 - Command line and centralized administrative operations	0.9.0.0, 0.10.0.0, 0.10.1.0
KIP-3 - Mirror Maker Enhancement	0.9.0.0
KIP-2 - Refactor brokers to allow listening on multiple ports and IPs	0.9.0.0
KIP-1 - Remove support of request.required.acks	0.9.0.0

KIPs under discussion

Please insert new rows in **sorted order** (ascending by KIP number). Please move the adopted/accepted KIPs to the "Adopted KIPs" table above.

KIP	Comment
KIP-59: Proposal for a kafka broker command	Sent emails to Dev discussion group. Work tracked under KAFKA-3663 .
KIP-125: ZookeeperConsumerConnector to KafkaConsumer Migration and Rollback	
KIP-135 : Send of null key to a compacted topic should throw non-retriable error back to user	
KIP 141 - ProducerRecord & SourceRecord: Add timestamp constructors	
KIP-142: Add ListTopicsRequest to efficiently list all the topics in a cluster	
KIP-148: Add a connect timeout for client	
KIP-166 - Add a tool to make amounts of replicas and leaders on brokers balanced	
KIP-169 - Lag-Aware Partition Assignment Strategy	
KIP-178: Size-based log directory selection strategy	
KIP-185: Make exactly once in order delivery the default producer setting	
KIP-193: Add SchemaBuilder.from(Schema)	
KIP-201: Rationalising Policy interfaces	
KIP-209 - Connection String Support	
KIP-217: Expose a timeout to allow an expired ZK session to be re-created	
KIP-228 Negative record timestamp support	Voting in progress
KIP-234: add support for getting topic defaults from AdminClient	
KIP-236: Interruptible Partition Reassignment	Discussion
KIP-240: AdminClient.listReassignments() AdminClient.describeReassignments()	

KIP-242: Mask password in Kafka Connect Rest API response	
KIP-250 Add Support for Quorum-based Producer Acknowledgment	
KIP-252 - Extend ACLs to allow filtering based on ip ranges and subnets	
KIP-253: Support in-order message delivery with partition expansion	Discussion
KIP-254: JsonConverter Exception Handling	
KIP-264: Add a consumer metric to record raw fetch size	Voting in progress
KIP-271: Add NetworkClient redirector	Discussion
KIP-273: Kafka to support using ETCD beside Zookeeper	Discussion
KIP-293 Add new metrics for consumer/replication fetch requests	Voting in progress
KIP-296: Add connector level configurability for producer/consumer client configs	Discussion
KIP-301: Schema Inferencing for JsonConverter	Discussion
KIP-304: Connect runtime mode improvements for container platforms	Discussion
KIP-315: Stream Join Sticky Assignor	Discussion
KIP-316: Command-line overrides for ConnectDistributed worker properties	Discussion
KIP-317: Add end-to-end data encryption functionality to Apache Kafka	Discussion
KIP-325: Extend Consumer Group Command to Show Beginning Offsets	Voting in Progress ³⁴⁵
KIP-333: Add faster mode of rebalancing	Discussion
KIP-334 - Include partitions in exceptions raised during consumer record deserialization/validation	
KIP-348 Eliminate null from SourceTask#poll()	
KIP-350: Allow kafka-topics.sh to take brokerid as parameter to show partitions associated with it	
KIP-356: Add KafkaConsumer fetch-error-rate and fetch-error-total metrics	Discussion
KIP-363: Allow performance tools to print final results to output file	Discussion
KIP-370: Remove Orphan Partitions	Discussion
KIP-375: Kafka Clients - make Metadata#TOPIC_EXPIRY_MS configurable	Discussion
KIP-381: Connect: Tell about records that had their offsets flushed in callback	Voting in progress (restarted 18th January 2019, due to no votes in first attempt)
KIP-384: Add config for incompatible changes to persistent metadata	Discussion
KIP-385: Avoid throwing away prefetched data	Discussion
KIP-387: Fair Message Consumption Across Partitions in KafkaConsumer	Discussion
KIP-388: Add observer interface to record request and response	Discussion
KIP-395: Encrypt-then-MAC Delegation token metadata	
KIP-400: Improve exit status in case of errors in ConsoleProducer	Discussion
KIP-403: Increase ProducerPerformance precision by using nanoTime	Draft
KIP-406: GlobalStreamThread should honor custom offset policy.	Blocked
KIP-407: Kafka Connect support override worker kafka api configuration with connector configuration that post by rest api	
KIP-409: Allow creating under-replicated topics and partitions	
KIP-410: Add metric for request handler thread pool utilization by request type	Discussion
KIP-416: Notify SourceTask of ACK'd offsets, metadata	Discussion
KIP-419: Safely notify Kafka Connect SourceTask is stopped	Voting in progress, JIRA exists with pull request
KIP-423: Add JoinReason to Consumer Join Group Protocol	Discussion
KIP-424: Allow suppression of intermediate events based on wall clock time	Discussion

KIP-426: Persist Broker Id to Zookeeper	Discussion
KIP-435: Internal Partition Reassignment Batching	Discussion
KIP-438: Expose task, connector IDs in Connect API	Discussion
KIP-439: Cleanup built-in Store interfaces	Discussion
KIP-448: Add State Stores Unit Test Support to Kafka Streams Test Utils	Discussion
KIP-452: Tool to view cluster status	Discussion, JIRA exists
KIP-468: Avoid decompression of record when validate record at server in the scene of inPlaceAssignment .	Under discussion
KIP-473: Enable KafkaLog4JAppender to use SASL Authentication Callback Handlers	Under Discussion
KIP-477: Add PATCH method for connector config in Connect REST API	Under Discussion
KIP-487: Automatic Topic Creation on Producer	Under Discussion
KIP-489: Kafka Consumer Record Latency Metric	Under Discussion
KIP-490: New metric to count offsets expired without being consumed by a consumer group	Under discussion
KIP-491: Preferred Leader Deprioritized List (Temporary Blacklist)	Under discussion
KIP-494: Connect REST Endpoint to get Transformations (SMTs)	
KIP-498: Add client-side configuration for maximum response size to protect against OOM	Sent emails to Dev discussion group. Work tracked under KAFKA-4090 .
KIP-502: Connect SinkTask.put(...) to specify ArrayList<SinkRecord> in Signature	
KIP-505: Add new public method to only update assignment metadata in consumer	Under discussion
KIP-506: Allow setting SCRAM password via Admin interface	Under discussion
KIP-508: Make Suppression State Queriable	Under discussion
KIP-509: Rebalance and restart Producers	Under discussion
KIP-510: Metrics library upgrade	Under discussion
KIP-512: Adding headers to RecordMetaData	Sent mail to dev discussion group
KIP-513: Distinguish between Key and Value serdes in scala wrapper library for kafka streams	Sent mail to dev discussion group
KIP-536: Propagate broker start time to Admin API	Under discussion
KIP-539: Add mechanism to flush records out in low volume suppression buffers	Under discussion.
KIP-540: Implement per key stream time tracking	Draft (In Progress)
KIP-542: Partition Reassignment Throttling	Draft
KIP-547: Extend ConsumerInterceptor to allow modification of Consumer Commits	Under Discussion
KIP-548 Add Option to enforce rack-aware custom partition reassignment execution	Under Discussion
KIP-549: Surface Consumer's Metadata in KafkaAdminClient#describeConsumerGroups	Under Discussion
KIP-550: Mechanism to Delete Stray Partitions on Broker	Draft
KIP-552: Suppress "unused config" warning	Under Discussion
KIP-560: Auto infer external topic partitions in stream reset tool	Under Discussion
KIP-561: Regex Support for ConsumerGroupCommand	Under Discussion
KIP-563: Add 'tail -n' feature for ConsoleConsumer	Under Discussion
KIP-564: Add new cached authorizer:change the dim of cache	Under Discussion
KIP-565: Using AclCommand,avoid call the global method loadcache in SimpleAclAuthorizer	Under Discussion
KIP-566: Add rebalance callbacks to ConsumerInterceptor	Under Discussion
KIP-567: Kafka Cluster Audit	Under Discussion
KIP-575: build a Kafka-Exporter by Java	Under Discussion

KIP-576: Support dynamic update of more broker configs related to replication	Under Discussion
KIP-578: Add configuration to limit number of partitions	Voting
KIP-579: new exception on min.insync.replicas > replication.factor	Voting in progress
KIP-582: Add a "continue" option for Kafka Connect error handling	Under Discussion
KIP-583: add tag "partition" to BrokerTopicMetrics so as to observe the partition metrics on the same broker	Under Discussion
KIP-587: Suppress detailed responses for handled exceptions in security-sensitive environments	Under Discussion
KIP-592: Replicate mirrmaker topics from earliest	Under Discussion
KIP-595: A Raft Protocol for the Metadata Quorum	Under Discussion
KIP-596: Safely abort Producer transactions during application shutdown	Under Discussion
KIP-598: Augment TopologyDescription with store and source / sink serde information	Under Discussion
KIP-619: Add internal topic creation support	Under Discussion
KIP-625: Richer encodings for integral-typed protocol fields	Under Discussion
KIP-628: ConsumerPerformance's multi-thread implementation	Sent email to Dev discussion group. Work tracked under KAFKA-10136 .
KIP-636: Make RPC error codes and messages tagged fields	Under Discussion
KIP-637: Include min.insync.replicas in MetadataResponse to make Producer smarter in partitioning events	Draft
KIP-639: Move nodeLevelSensor and storeLevelSensor methods from StreamsMetricsImpl to StreamsMetrics	Under Discussion
KIP-640: Add log compression analysis tool	Under Discussion
KIP-649: Dynamic client configuration	Under Discussion
KIP-655: Windowed Distinct Operation for Kafka Streams API	Voting in progress
KIP-656: MirrorMaker2 Exactly-once Semantics	Draft
KIP-665: Kafka Connect Hash SMT	Voting in progress
KIP-668: Expose REST endpoint to list converter plugins	Under Discussion
KIP-669 Preserve Source Partition in Kafka Streams from context	Draft
KIP-674: API to Aggregate Metrics in Kafka Streams	Under Discussion
KIP-675: Convert a KTable to a KStream using the previous value	Under Discussion
KIP-678: New Kafka Connect SMT for plainText => struct with Regex	Voting
KIP-681: Rename master key in delegation token feature	Draft
KIP-682: Connect TimestampConverter support for multiple fields and multiple input formats	Under Discussion
KIP-683: Add recursive support to Connect Cast and ReplaceField transforms, and support for casting complex types to either a native or JSON string	Under Discussion
KIP-685: Loosen permission for listing reassignments	Under Discussion
KIP-686: API to ensure Records policy on the broker	Under Discussion
KIP-687: Automatic Reloading of Security Store	Under Discussion
KIP-688: Support dynamic update of delete.topic.enable config	Under Discussion
KIP-692: Make AdminClient value object constructors public	Under discussion
KIP-693: Client-side Circuit Breaker for Partition Write Errors	Voting in progress
KIP-694: Support Reducing Partitions for Topics	Under Discussion
KIP-697: Stricter parsing of addresses in configs	Under Discussion
KIP-701: Add --override option to bin/connect-distributed.sh	Under Discussion

KIP-702: The control plane needs to force the validation of requests from the controller	Under Discussion
KIP-703: Add a metric for reporting idle connections closed	Under Discussion
KIP-705: Selectively Disable Topology Optimizations	Draft
KIP-706: Add method "Producer#produce" to return CompletionStage instead of Future	Under Discussion
KIP-711: Deprecate org.apache.kafka.streams.errors.BrokerNotFoundException	Under Discussion
KIP-712: Shallow Mirroring	Under Discussion
KIP-713: Validation of Enums in configuration	Under Discussion
KIP-718: Make KTable Join on Foreign key unopinionated	Under Discussion
KIP-723: Add socket.tcp.no.delay property to Kafka Config	Under Discussion
KIP-727: Add --under-preferred-replica-partitions option to describe topics command	Under Discussion
KIP-729: Custom validation of records on the broker prior to log append	Under Discussion
KIP-731: Record Rate Limiting for Kafka Connect	Under Discussion
KIP-736: Report the true end to end fetch latency	Under Discussion
KIP-737: Add canTrackSource to ReplicationPolicy	Under Discussion
KIP-739: Block Less on KafkaProducer#send	Under Discussion
KIP-749: Add --files and --file-separator options to the ConsoleProducer	Under Discussion
KIP-752: Support --bootstrap-server in ReplicaVerificationTool	Under Discussion
KIP-754: Make Scala case class's final	Under Discussion
KIP-755: Add new AUTO_CREATE ACL for auto topic creation	Under Discussion
KIP-759: Unneeded repartition canceling	Accepted, targeting 3.7
KIP-760: Minimum value for segment.ms and segment.bytes	Under Discussion
KIP-762: Delete Committed Connect Records	Under Discussion
KIP-767: Connect Latency Metrics	Under Discussion
KIP-774: Deprecate public access to Admin client's *Result constructors	Under Discussion
KIP-777: Improved testability for Admin client	Draft
KIP-780: Support fine-grained compression options	Under Discussion
KIP-781: Improve MirrorMaker2's client configuration	Under Discussion
KIP-782: Expandable batch size in producer	Voting
KIP-785: Automatic storage formatting	Under Discussion
KIP-786: Emit Metric Client Quota Values	Under Discussion
KIP-789: Use localhost:9092 as default bootstrap-server/broker-list in client tools	KIP page WIP
KIP-799: Align behaviour for producer callbacks with documented behaviour	Under Discussion
KIP-802: Validation Support for Kafka Connect SMT Options	Under Discussion
KIP-803: Add Task ID and Connector Name to Connect Task Context	Under Discussion
KIP-804: OfflinePartitionsCount Tagged by Topic	Under Discussion
KIP-807: Refactor KafkaStreams exposed metadata hierarchy	Under Discussion
KIP-809: Support live upgrades with dynamic addition/removal of modular topologies	KIP page WIP
KIP-816: Topology changes without local state reset	Under Discussion
KIP-817: Fix inconsistency in dynamic application log levels	Under Discussion
KIP-819: Merge multiple KStreams in one operation	Under Discussion

KIP-822: Optimize the semantics of <code>KafkaConsumer#pause</code> to be consistent between the two <code>RebalanceProtocols</code>	Under Discussion
KIP-823: Update <code>Admin::describeConfigs</code> to allow fetching specific configurations	KIP page WIP
KIP-826: Define platforms supported	Under Discussion
KIP-828: Add the corresponding validator to the configuration where the validator is missing	Under Discussion
KIP-829: (console-consumer) add <code>print.topic</code> property	Under Discussion
KIP-838 Simulate batching and compression	Under Discussion
KIP-839: Provide builders for <code>KafkaProducer/KafkaConsumer</code> and <code>KafkaStreams</code>	Under Discussion
KIP-842: Add richer group offset reset mechanisms	Under Discussion
KIP-849: Expose logdirs total and usable space via <code>kafka-log-dirs.sh</code>	Under Discussion
KIP-853: KRaft Voter Changes	Under Discussion
KIP-855: Add <code>schema.namespace</code> parameter to <code>SetSchemaMetadata SMT</code> in Kafka Connect	Under Discussion
KIP-856: KRaft Disk Failure Recovery	Draft
KIP-857: Streaming recursion in Kafka Streams	Under Discussion
KIP-860: Add client-provided option to guard against replication factor change during partition reassignments	Under Discussion
KIP-864: Add End-To-End Latency Metrics to Connectors	Under Discussion
KIP-870: Retention policy based on record event time	Under Discussion
KIP-873: <code>ExceptionHandlingDeserializer</code> , <code>RetryDeserializer</code> , <code>PipeSerializer</code> , <code>PipeDeserializer</code>	Under Discussion
KIP-877: Mechanism for plugins and connectors to register metrics	Under Discussion
KIP-879: Multi-level Rack Awareness	Under Discussion
KIP-880: X509 SAN based SPIFFE URI ACL within mTLS Client Certificates	Under Discussion
KIP-882: Kafka Connect REST API configuration validation timeout improvements	Under Discussion
KIP-883: Add <code>isDeleted</code> flag when stopping a connector	Under Discussion
KIP-885: Expose Broker's Name and Version to Clients	Under Discussion
KIP-886: Add Client Producer and Consumer Builders	Under Discussion
KIP-888: Batch describe ACLs and describe client quotas	Under Discussion
KIP-891: Running multiple versions of a connector	Under Discussion
KIP-895: Dynamically refresh partition count of <code>__consumer__offsets</code>	Under Discussion
KIP-899: Allow clients to bootstrap	Under Discussion
KIP-901: Add <code>connectorDeleted</code> flag when stopping tasks	Under Discussion
KIP-905: Broker interceptors	Under Discussion
KIP-910: Update Source offsets for Source Connectors without producing records	Under Discussion
KIP-912: Support decreasing send's block time without worrying about metadata's fetch	Under Discussion
KIP-913: add new method to provide possibility for accelerate first record's sending	Under Discussion
KIP-916: MM2 distributed mode flow log context	Under Discussion
KIP-917: Additional custom metadata for remote log segment	Under Discussion
KIP-918: MM2 Topic And Group Listener	Under Discussion
KIP-921: OpenJDK CRaC support	Under Discussion
KIP-922: Add the traffic metric of the partition dimension	Under Discussion
KIP-924: Introduce public config for Streams task assignor	Under Discussion
KIP-926: introducing <code>acks=min.insync.replicas</code> config	Under Discussion

KIP-928: Making Kafka resilient to log directories becoming full	Under Discussion
KIP-930: Rename ambiguous Tiered Storage Metrics	Under Discussion
KIP-931: Flag to ignore unused message attribute field	Under Discussion
KIP-932: Queues for Kafka	Under Discussion
KIP-933: Publish metrics when source connector fails to poll data	Under Discussion
KIP-934: Add DeleteTopicPolicy	Under Discussion
KIP-935: Extend AlterConfigPolicy with existing configurations	Under Discussion
KIP-936: Throttle number of active PIDs	Under Discussion
KIP-939: Support Participation in 2PC	Under Discussion
KIP-940: Broker extension point for validating record contents at produce time	Under Discussion
KIP-943: Add independent "offset.storage.segment.bytes" for connect-distributed.properties	Under Discussion
KIP-945: Update threading model for Consumer	WIP
KIP-946: Modify exceptions thrown by Consumer APIs	WIP
KIP-948: Allow custom prefix for internal topic names in Kafka Streams	Under Discussion
KIP-950: Tiered Storage Disablement	Under Discussion
KIP-952: Regenerate segment-aligned producer snapshots when upgrading to a Kafka version supporting Tiered Storage	Under Discussion
KIP-953: partition method to be overloaded to accept headers as well.	Draft
KIP-955: Add stream-table join on foreign key	Under Discussion
KIP-956 Tiered Storage Quotas	Draft
KIP-964 Have visibility when produce requests become "async"	Under discussion
KIP-965: Support disaster recovery between clusters by MirrorMaker	Under Discussion
KIP-969: Support range Interactive Queries (IQv2) for Versioned State Stores	
KIP-971: Expose replication-record-lag MirrorMaker2 metric	Voting in progress
KIP-972: Add the metric of the current running version of kafka	Under Discussion
KIP-973: Expose per topic replication rate metrics	Under Discussion
KIP-974: Docker Image for GraalVM based Native Kafka Broker	Under Discussion
KIP-981: Manage Connect topics with custom implementation of Admin	Under Discussion
KIP-982: Access SslPrincipalMapper and kerberosShortNamer in Custom KafkaPrincipalBuilder	Voting in progress (PR is reviewed by Contributors)
KIP-983: Full speed async processing during rebalance	Draft
KIP-984: Add pluggable compression interface to Kafka	Under Discussion
KIP-986: Cross-Cluster Replication	Draft
KIP-987: Connect Static Assignments	Under Discussion
KIP-989: RocksDB Iterator Metrics	Under Discussion
KIP-990: Capability to PAUSE Tasks on DeserializationException	Under Discussion
KIP-991: Allow DropHeaders SMT to drop headers by wildcard/regexp	Under Discussion
KIP-995: Allow users to specify initial offsets while creating connectors	Under Discussion
KIP-997: update WindowRangeQuery and unify WindowKeyQuery and WindowRangeQuery	Under Discussion Vote
KIP-999: Server-side Consumer Lag Metrics	Draft
KIP-1002: Fetch remote segment indexes at once	Under Discussion

KIP-1003: Signal next segment when remote fetching	Under Discussion
KIP-1006: Remove SecurityManager Support	Draft
KIP-1008: ParKa - the Marriage of Parquet and Kafka	Under Discussion
KIP-1009: Add Broker-level Throttle Configurations	Under Discussion
KIP-1010: Topic Partition Quota	Under Discussion
KIP-1011: Use incrementalAlterConfigs when updating broker configs by kafka-configs.sh	Under Discussion
KIP-1015: Limit number of ssl connections in brokers	Under Discussion
KIP-1016 Make MM2 heartbeats topic name configurable	Under Discussion
KIP-1017: Health check endpoint for Kafka Connect	Under Discussion
KIP-1018: Introduce max remote fetch timeout config for DelayedRemoteFetch requests	Under Discussion
KIP-1019: Expose method to determine Metric Measurability	Under Discussion

[Go to link](#)

Dormant/inactive KIPs

Please insert new rows in **sorted order** (ascending by KIP number).

KIP	Comment
KIP-6 - New reassignment partition logic for rebalancing	Needs more detail
KIP-10 - Running Producer, Consumers and Brokers on Mesos	
KIP-14 - Tools standardization	
KIP-17 - Add HighwaterMarkOffset to OffsetFetchResponse	
KIP-23 - Add JSON/CSV output and looping options to ConsumerGroupCommand	
KIP-27 - Conditional Publish	
KIP-30 - Allow for brokers to have plug-able consensus and meta data storage sub systems	
KIP-37 - Add Namespaces to Kafka	
KIP-39: Pinning controller to broker	
KIP-44 - Allow Kafka to have a customized security protocol	
KIP-46 - Self Healing	
KIP-47 - Add timestamp-based log deletion policy	
KIP-49 - Fair Partition Assignment Strategy	
KIP-53 - Add custom policies for reconnect attempts to NetworkdClient	
KIP-59: Proposal for a kafka broker command	
KIP-64 -Allow underlying distributed filesystem to take over replication depending on configuration	
KIP-68 Add a consumed log retention before log retention	
KIP-69 - Kafka Schema Registry	Draft
KIP-76 Enable getting password from executable rather than passing as plaintext in config files	
KIP-87 - Add Compaction Tombstone Flag	Possibly displaced by KIP-82
KIP-95: Incremental Batch Processing for Kafka Streams	
KIP-159: Introducing Rich functions to Streams	

KIP-260: add primary join operation for Stream-Stream join (WIP)	Draft
KIP-269: Substitution Within Configuration Values	
KIP-311: Async processing with dynamic scheduling in Kafka Streams	
KIP-314: KTable to GlobalKTable Bi-directional Join	
KIP-326: Schedulable KTable as Graph source	
KIP-349: Priorities for Source Topics	Functionality can be realized with existing API although at a lower level.
KIP-335: Consider configurations for KafkaStreams	
KIP-362: Support Dynamic Session Window	
KIP-378: Enable Dependency Injection for Kafka Streams handlers	
KIP-408: Add asynchronous processing to Kafka Streams	
KIP-422: Add support for client quota configuration in the Kafka Admin Client	KIP-546 added the necessary APIs instead
KIP-457: Add DISCONNECTED status to Kafka Streams	
KIP-459: Improve KafkaStreams#close	
KIP-463: Auto-configure non-default Serdes passed alongside the TopologyBuilder	
KIP-611: Improved Handling of Abandoned Connectors and Tasks	
KIP-645: Replace Windows with a proper interface	Still nice to have, just not as important as I thought.
KIP-634: Complementary support for headers and record metadata in Kafka Streams DSL	KIP-820 offers a workaround, and potential DSL v2 could replace the proposal.

Discarded KIPs

Please insert new rows in **sorted order** (ascending by KIP number).

KIP	Comment
KIP-5 - Broker Configuration Management	Superseded by KIP-21
KIP-7 - Security - IP Filtering	
KIP-9 - SSL Support	Adopted via KAFKA-1690 but not via this KIP process
KIP-18 - JBOD Support	Superseded by KIP-112 and KIP-113
KIP-24 - Remove ISR information from TopicMetadataRequest and add broker level metadata request	
KIP-29 - Add an IsrPropagateIntervalMs configuration to KafkaConfig	No longer need after KAFKA-2722
KIP-34 Add Partitioner Change Listener to Partitioner Interface for Multiple Use Case	Per feedback it is better do this partition() method and avoid Thread Coordination etc.
KIP-50 - Move Authorizer to o.a.k.common package	Superseded by KIP-504
KIP-80: Kafka Rest Server	
KIP-83 - Allow multiple SASL authenticated Java clients in a single JVM process	After KIP-85, no interface changes, reduced to KAFKA-4180
KIP-111: Kafka should preserve the Principal generated by the PrincipalBuilder while processing the request received on socket channel, on the broker.	Covered by KIP-189
KIP-116: Add State Store Checkpoint Interval Configuration	Currently not needed as checkpointing can be done on commit interval.
KIP-127: Pluggable JAAS LoginModule configuration for SSL	Similar functionality can be implemented using a custom PrincipalBuilder.
KIP-132: Augment KStream.print to allow extra parameters in the printed string	Duplicated by KIP-160
KIP-147: Add missing type parameters to StateStoreSupplier factories and KGroupedStream/Table methods	Covered by KIP-182

KIP-165: Extend Interactive Queries for return latest update timestamp per key	Covered by KIP-258
KIP-170: Enhanced TopicCreatePolicy and introduction of TopicDeletePolicy	Superseded by KIP-201: Rationalising Policy interfaces
KIP 172: Add regular-expression topic support for sink connector	Superseded by KIP-215: Add topic regex support for Connect sinks
KIP-179: Change ReassignPartitionsCommand to use AdminClient	
KIP-184: Rename LogCleaner and related classes to LogCompactor	A lot of configuration change will cause more trouble and doesn't seems to be worth it
KIP-199: Add Kafka Connect offset tool	Superseded by KIP-875: First-class offsets support in Kafka Connect
KIP 230: Name Windowing Joins	Covered by KIP-372: Naming Repartition Topics for Joins and Grouping
KIP-232: Detect outdated metadata using per-partition leaderEpoch field	Merged into KIP-320: Allow fetchers to detect and handle log truncation
KIP-246: Connect producers and consumers should inherit worker configs	Withdrawn because proposal could not be made backward compatible with existing behavior
KIP-248 - Create New ConfigCommand That Uses The New AdminClient	Discarded by author as it's partly covered in KIP-339 and KIP-422
KIP-259: Improve Streams DSL Timestamp Propagation Semantics	Change did not require a KIP.
KIP-262: Metadata should include number of state stores for task	Rejected since metadata is no longer required.
KIP-263: Allow broker to skip sanity check of inactive segments on broker startup	Withdrawn because the solution we agreed on does not require interface change
KIP-275 - Indicate "isClosing" in the SinkTaskContext	
KIP-286: producer.send() should not block on metadata update	Withdrawn because the benefit of not having to wait for metadata is probably not worth the complexity added in producer.
KIP-288: [DISCARDED] Consumer.poll() timeout semantic change and new waitForAssignment method	Discarded in deference to KIP-266
KIP-310: Add a Kafka Source Connector to Kafka Connect	Withdrawn in favor of KIP-382.
KIP-327: Add describe all topics API to AdminClient	Withdrawn in-favor of filter support in Metadata API and KIP-142
KIP-344: The auto-generated client id should be passed to MetricsReporter	Bug fix so KIP not needed
KIP-347: Enable batching in FindCoordinatorRequest	Replaced by KIP-699
KIP-364: Remove implicit Materialized, Consumed and Produced	Subsumed by KIP-365
KIP-383: Pluggable interface for SSL Factory	Replaced by KIP-519
KIP-391: Allow Producing with Offsets for Cluster Replication	Not enough interest in favour.
KIP-398: Support reading trust store from classpath	Not enough interest in favour. Can be implemented with mechanism in  Unable to render Jira issues macro, execution error.
KIP-404: Add Kafka Connect configuration parameter for disabling WADL output on OPTIONS request	Discarded, reported as a bug by  Unable to render Jira issues macro, execution error.
KIP-432: Additional Broker-Side Opt-In for Default, Unsecure SASL /OAUTHBEARER Implementation	A "security sanity check tool" is a more generic and appropriate solution.
KIP-451: Make TopologyTestDriver output iterable	Discarded in favor of KIP-456
KIP-456: Helper classes to make it simpler to write test logic with TopologyTestDriver	Discarded in favor of KIP-470

KIP-472: Add header to RecordContext/ProducerRecord	Discarded due to change being far more complex than necessary for a simple bug
KIP-485: Make topic optional when using through() operations in DSL	Moved to KIP-221
KIP-486: Support custom way to load KeyStore and TrustStore	Discarded in favor of KIP-519
KIP-522: Update BrokerApiVersionsCommand to use AdminClient	Discarded as this KIP is likely to be misused
KIP-593: Enable --if-exists and --if-not-exists for AdminClient in TopicCommand	Superseded by KIP-604
KIP-615: add ConstrainedCooperativeStickyAssignor	Discarded in favor of modifying existing assignor
KIP-638: Deprecate DescribeLogDirsResponse.[LogDirInfo, ReplicaInfo]	Duplication of KIP-621
KIP-660: Pluggable ReplicaPlacer	
KIP-667: Remove deprecated methods from ReadOnlyWindowStore	No KIP needed
KIP-672: Introduce Kafka Streams Specific Uncaught Exception Handler	Combined with KIP-671
[DISCARD] KIP-658 Deprecate all setters of Headers	
KIP-818: Introduce cache-size-bytes-total Task Level Metric	Decided to add the new metric in KIP-770.
KIP-717: Deprecate batch-size config from console producer	Change did not require a KIP
KIP-756: Move StreamsResetter tool outside of core	Subsumed by KIP-906
KIP-795: Add public APIs for AbstractCoordinator	Discarded in favor of KIP-848
KIP-832: Allow creating a producer/consumer using a producer/consumer config	Superseded by KIP-839
KIP-845: 'HasField' predicate for kafka connect	Discarded in favor of improving existing SMTs
KIP 850: REST API for filtering Connector plugins by type	Discarded as JIRA was obsolete
KIP-871: Fix ByteBufferSerializer#serialize(String, ByteBuffer) compatible problem	Discard as it doesn't change any public API.
KIP-874: TopicRoundRobinAssignor	Discard as it is too specific.
KIP-897: Publish a single kafka (aka core) Maven artifact in Apache Kafka 4.0	Taking a different approach.
KIP-908: Add description field to connector configuration	Discarded as the benefits are too limited
KIP-947: Add support for consumer group protocol in Consumer	Change did not require a KIP
KIP-944: Support async runtimes in consumer and KIP-957: Support Async runtimes	Withdrawn because the committers do not seem to be convinced that you cannot control on what thread code runs with an async runtime

KIP Discussion Recordings

Date (link to recording)	Summary
2018-04-09	KIP-253 - partition expansion: We discussed a few things. (1) Is it useful to backfill a compacted topic? The main use case is to rebuild the application states. If the new partition has the existing data, rebuilding the state can be done easily by reading from a single partition. Otherwise, an application has to read both the child and the parent partition to rebuild the state. This is possible, but can be complicated. Jan will do an exercise to see how complicated this is. (2) What's the best way to add the backfilling support if we want to do it? We can do this on the server side or on the client side. The former potentially makes the coordination easier. The latter potentially reduces the memory footprint on the server for reshuffling. We need to think through how to support EOS message format and how to throttle the process to avoid overwhelming the cluster. (3) Linear hashing vs doubling partitions. It seems that Linear hashing is more general. (4) Partition splitting in Kinesis. This is done differently since it doesn't allow customized partitioning. It doesn't support compacted topic either. (5) Sticky partition assignment. It could be useful to support a partition assignment strategy where the child partition is assigned together with the parent partition to a consumer instance so that the local state doesn't have to be moved immediately. (6) Consumer callback on partition splitting. This could still be useful if the states are maintained globally.

2017-01-07	KIP-112 - Handle disk failure for JBOD: We discussed whether we need to support JBOD directly in Kafka or just rely on the 1 disk per broker model. The general consensus is that direct JBOD support in Kafka is needed. There is some concern on the complexity added to Kafka. So, we have to be careful with the implementation details. We discussed how directory failure should be detected, where the failure state is kept, and whether the state should be reset on broker restart. There is a bit confusing on what's written in the wiki. Dong is going to clarify the proposal based on the feedback and we will follow up on the details in the mailing list.
2016-10-19	KIP-82 - add record header: We agreed that there are use cases for third-party vendors building tools around Kafka. We haven't reached the conclusion whether the added complexity justifies the use cases. We will follow up on the mailing list with use cases, container format people have been using, and details on the proposal.
2016-09-13	- KIP-54 (Sticky Partition Assignment): aims to minimise partition movement so that resource reinitialisation (e.g. caches) is minimised. It is partially sticky and partially fair. Some concerns around the fact that user code for partitionsRevoked and partitionsAssigned would have to be changed to work correctly with this assignment strategy. Good: more complex usage of an assigner that takes advantage of the user data field. Vahid will start the vote. - KIP-72 (Allow Sizing Incoming Request Queue in Bytes): large requests can kill the broker, no control over how much memory is allocated. Client quotas don't help as damage may already have been done by the time they kick in. There was a discussion on whether it was worth it to avoid the immediate return from select when there was no memory available in the pool. Radai will update the KIP to describe this aspect in more detail as well as the config validation that is performed. - KIP-79 (ListOffsetRequest/ListOffsetResponse v1 and add timestamp search methods to the new consumer): we discussed the option of passing multiple timestamps for the same partition in the same request. Becket thinks it's a rare use case and not worth supporting. Gwen said that it would be nice to have, but not essential. We talked about validation of duplicate topics. Becket will check the approach taken by the create topics request and evaluate if it can be adopted here too. PR will be available today and Jason will evaluate if it's feasible to include it in the next release once it's available.
2016-08-30	- KIP-48 (delegation tokens): Harsha will update the wiki with more details on how to use delegation tokens and how to configure it. - KIP-78 (cluster id): There was discussion on adding human readable tags later. No major concerns.
2016-08-23	- time-based release: No one seems to have objections. Ismael will follow up with a release wiki. - KIP-4: We discussed having separate ACL requests of add and delete. No one seems to object to it. We discussed the admin client. Grant will send a PR. We discussed how KStream can use the ACL api. It seems that we will need some kind of regex or namespace support in ACL to make the authorization convenient in KStream. - KIP-50: There is some discussion for further changes in the PR. Ashish will reply to the KIP email thread with the recommended changes. Ashish/Grant plan to look into whether it's possible to make the authorizer api change backward compatible. However, it seems that people are in general ok with a non-compatible api change. - KIP-74: No objections on the current proposal. - Java 7 support timeline: The consensus is to defer dropping the Java 7 support until the next major release (which will be next year). Ismael will follow up on the email thread. - KIP-48 delegation token : Ashish will ping Harsh to see if this is still active. - Some of the KIPs have been idle. Grant will send a proposal on tagging them properly (e.g., blocked, inactive, no resource, etc).
2016-05-24	- KIP-58 - Make Log Compaction Point Configurable: We want to start with just a time-based configuration since there is no good usage for byte-based or message-based configuration. Eric will change the KIP and start the vote. - KIP-4 - Admin api: Grant will pick up the work. Initially, he plans to route the write requests from the admin clients to the controller directly to avoid having the broker forward the requests to the controller. - KIP-48 - Delegation tokens: Two of the remaining issues are (1) how to store the delegation tokens and (2) how token expiration works. Since Parth wasn't able to attend the meeting. We will follow up in the mailing list.
2016-04-05	- KIP-4: There is a slight debate on the metadata request schema, as well as the internal ZK based implementation, which we will wait for Jun to comment on the mailing list thread. - KIP-52: We decided to start a voting process for this. - KIP-35: Decided on renaming ApiVersionQuery api to ApiVersion. Consensus on using the api in java client to only check for availability of current versions. ApiVersion api's versions will not be deprecated. Update KIP-35 wiki will be updated with latest info and vote thread will be initiated.

2016-03-15	- KIP-33 - Add a time based log index to Kafka: We decided NOT to include this in 0.10.0 since the changes may have performance risks. - KIP-45 - Standardize all client sequence interaction on j.u.Collection: There is no consensus in the discussion. We will just put it to vote. - KIP-35 - Retrieving protocol version: This gets the longest discussion. There is still no consensus. Magnus thinks the current proposal of maintaining a global protocol version won't work and will try to submit a new proposal. - KIP-43 - Kafka SASL enhancements: Rajini will modify the KIP to only support native SASL mechanisms and leave the changes to Login and CallbackHandler to KIP-44 instead.
2016-02-23	- KIP-33 and KIP-47: No issues. Will start the voting thread. - KIP-43: We discussed whether there is a need to support multiple SASL mechanisms at the same time and what's the best way to implement this. Will discuss this in more details in the email thread. - KIP-4: Grant gave a comprehensive summary of the current state. We have gaps on how to make the admin request block on the broker, how to integrate admin requests with ACL (especially with respect to client config changes for throttling and ACL changes), how to do the alter topic request properly. Grant will update the KIP with an interim plan and a long term plan. - KIP-43: We briefly discussed on to support multiple sasl mechanisms on the broker. Harsha will follow up with more details on the email thread. - Everyone seems to be in favor of making the next major release 0.10.0, instead of 0.9.1.
2016-01-26	- KIP-42: We agreed to leave the broker side interceptor for another KIP. On the client side, people favor the 2nd option in Anna's proposal. Anna will update the wiki accordingly. - KIP-43: We discussed whether there is a need to support multiple SASL mechanisms at the same time and what's the best way to implement this. Will discuss this in more details in the email thread. - Jiangjie brought up an issue related to KIP-32 (adding timestamp field in the message). The issue is that currently there is no convenient way for the consumer to tell whether the timestamp in a message is the create time or the server time. He and Guozhang propose to use a bit in the message attribute to do that. Jiangjie will describe the proposal in the email thread.
2016-01-12	- KIP-41: Discussed whether the issue of long processing time between poll calls is a common issue and whether we should revisit the poll api. Also discussed whether the number of records returned in poll calls can be made more dynamic. In the end, we feel that just adding a config that controls the number records returned in poll() is the simplest approach at this moment. - KIP-36: Need to look into how to change the broker JSON representation in ZK w/o breaking rolling upgrades. Otherwise, ready for voting.
2015-10-20	- KIP-38: No concerns with this KIP. Flavio will initiate the voting on this. - KIP-37: There are questions on how ACL, configurations, etc will work, and whether we should support "move" or not. We will discuss the details more in the mailing list. - KIP-32/KIP-33: Jiangjie raised some concerns on the approach that Jay proposed. Guozhang and Jay will follow up on the mailing list.
2015-10-13	0.9.0 release: We discussed if KAFKA-2397 should be a blocker in 0.9.0. Jason and Guozhang will follow up on the jira. - KIP-32 and KIP-33: We discussed Jay's alternative proposal of just keeping CreateTime in the message and having a config to control how far off the CreateTime can be from the broker time. We will think a bit more on this and Jiangjie will update the KIP wiki. - KIP-36: We discussed an alternative approach of introducing a new broker property to designate the rack. It's simpler and potentially can work in the case when the broker to rack mapping is maintaining externally. We need to make sure that we have an upgrade plan for this change. Allen will update the KIP wiki
2015-10-06	We only had the time to go through KIP-35. The consensus is that we will add a BrokerProtocolRequest that returns the supported versions for every type of requests. It's up to the client to decide how to use this. Magnus will update the KIP wiki with more details.
2015-09-22	- KIP-31: Need to figure out how to evolve inter.broker.protocol.version with multiple protocol changes within the same release, mostly for people who are deploying from trunk. Becket will update the wiki. - KIP-32/KIP-33: Having both CreateTime and LogAppendTime per message adds significant overtime. There are a couple of possibilities to improve this. Becket will follow up on this. - LinkedIn has been testing SSL in MirrorMaker (SSL is only enabled in the producer). So far, MirrorMaker can keep up with the load. LinkedIn folks will share some of the performance results.
2015-09-14	- KIP-28: Discussed the improved proposal including 2 layers of API (the higher layer is for streaming DSL), and stream time vs processor time. Ready for review. - KIP-31, KIP-32: (1) Discussed whether the timestamp should be from the client or the broker. (2) Discussed the migration path and whether this requires all consumers to upgrade before the new message format can be used. (3) Since this is too big a change, it will NOT be included in 0.9.0 release. Becket will update the wiki.

2015-08-18	• client-side assignment strategy: We discussed concerns about rebalancing time due to metadata inconsistency, especially when lots of topics are subscribed. Will discuss a bit more on the mailing list. • CopyCat data api: The discussions are in KAFKA-2367 for people who are interested. • 0.8.2.2: We want to make this a low risk bug fix release since 0.8.3 is coming. So, will only include a small number of critical and small fixes. • 0.8.3: The main features will be security and the new consumer. We will be cutting a release branch when the major pieces for these new features have been committed.
2015-08-11	• KIP-29: we will do a quick fix for unblocking production issues with hard-coded interval values, still needs further discussion on KIP itself. • KIP-26: RP-99 (https://github.com/apache/kafka/pull/99) pending for reviews. • KIP-28: RP-130 (https://github.com/apache/kafka/pull/130) needs further discussion. • KIP-4: KAFKA-1695 / 2210 pending for reviews.